Linking a SAP Build app to Microsoft Teams

I have been investigating easy ways to demonstrate the capabilities of Looply, the SAP:Microsoft Teams integration platform.

One great feature of Looply is the lo-code / no-code approach to building Teams integrations, which makes it a fantastic tool for citizen developers.

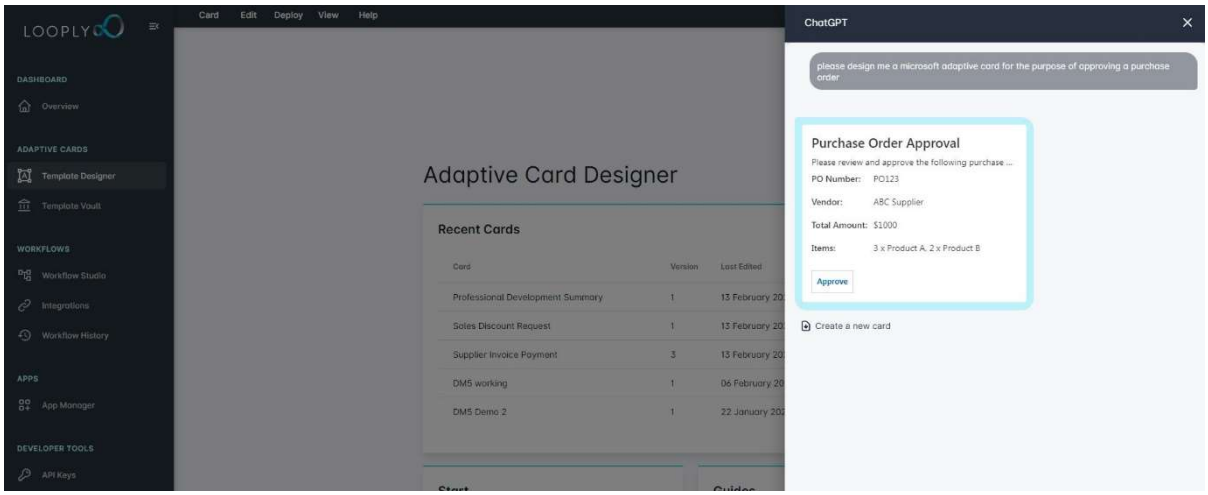
So, since SAP Build is the cornerstone of SAP's citizen developer offering, I decided to try to build a simple app using SAP Build, and see what steps are required for the app to trigger the creation of adaptive cards within Teams.

Like everything, as I was starting with zero knowledge of SAP Build and little knowledge of Looply, the first time involved a lot of learning. But with a little experience of both, the integration turns out to be ridiculously simple.

Building demo cards

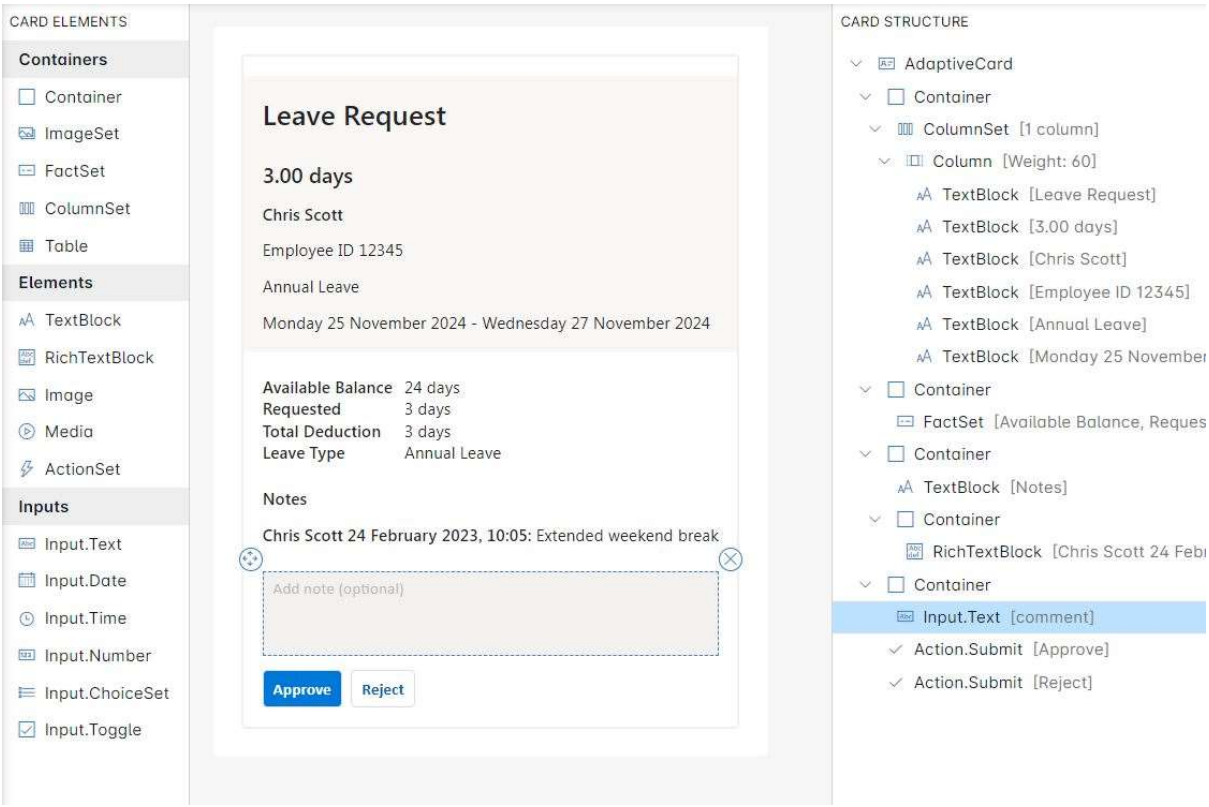
I didn't set off with a specific business process to model, but instead decided to build a series of cards with hard-coded data to simply illustrate what could be possible.

So without any back-end process to worry about, I could immediately start building some sample cards within Looply.



The option to use AI to build a card proved very useful. Basically there's an in-built integration with ChatGPT that enables you to define your requirements and a card is generated for you.

I could then edit the generated card using the Template Designer, and then create a copy, change all the fields and re-save.



In this way I was able to create a few different example cards very quickly, as of course there was no data binding or dynamic logic to consider.

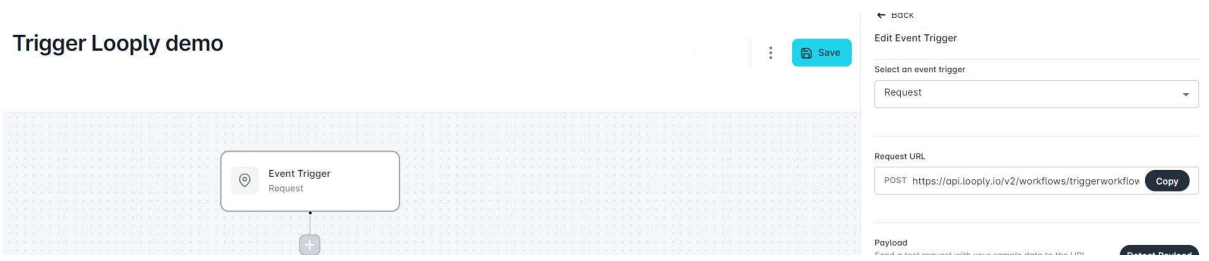
Building the Looply logic

The Looply logic takes the form of a Workflow, which is built using the Workflow Studio.

The Workflow needs to be built alongside the SAP Build app. While not necessarily an iterative process, the key steps need to be:

1) Begin the Looply Workflow build, and save to create the API 'endpoint'
2) Build the Looply app, and trigger the Looply Workflow from the app in test mode, to pass in the data model
3) Complete the build of the Looply Workflow, now with visibility of the data model.

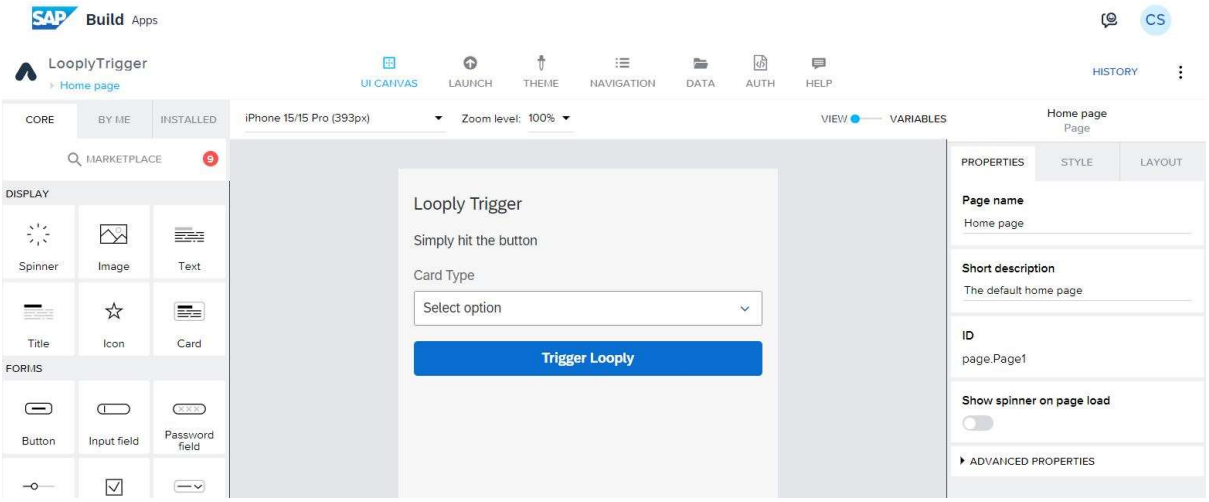
Starting the Workflow build means literally giving the workflow a name, selecting the 'event trigger' starting point, and hitting Save.



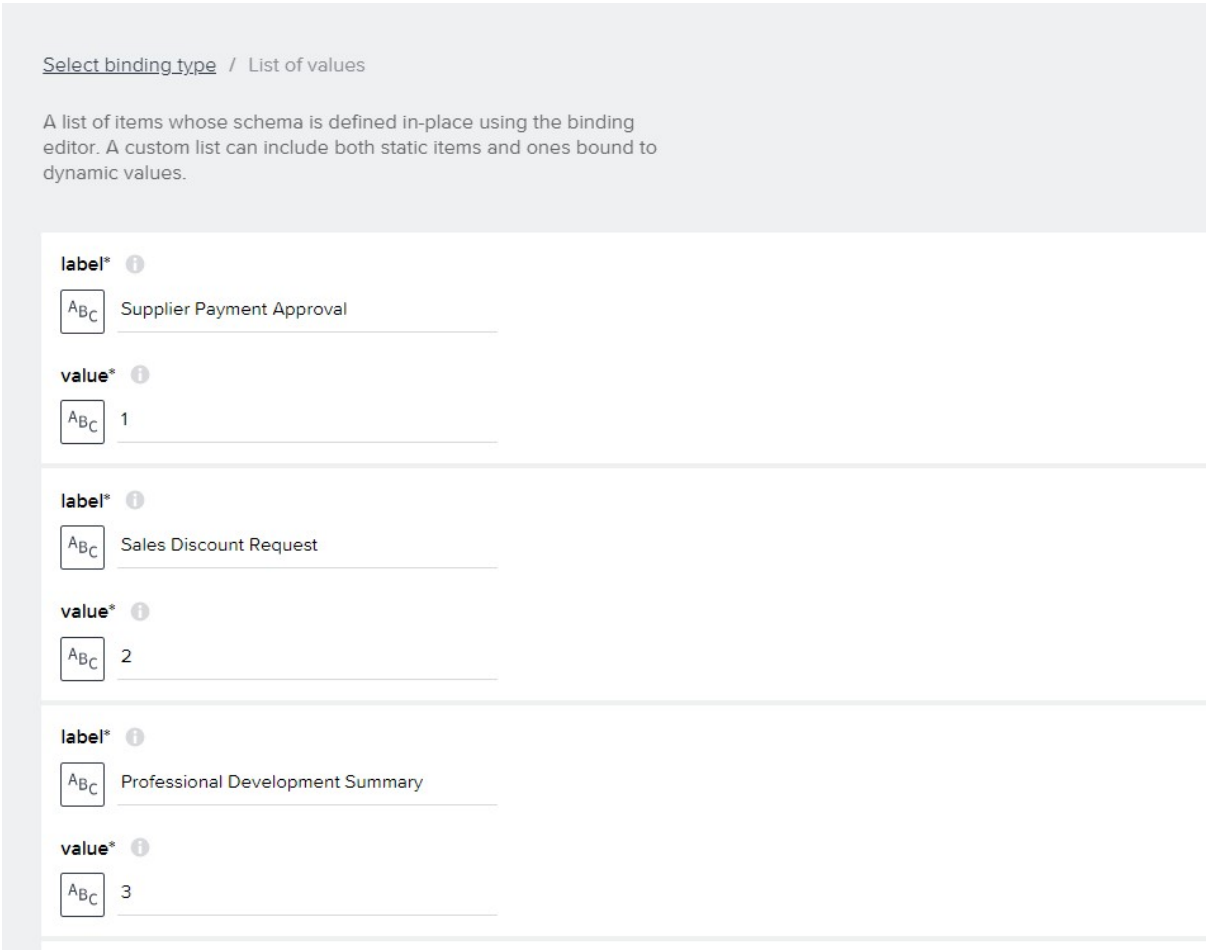
This generates a 'Request URL' that you can copy, as you will need this in your SAP Build app.

Building the SAP Build app

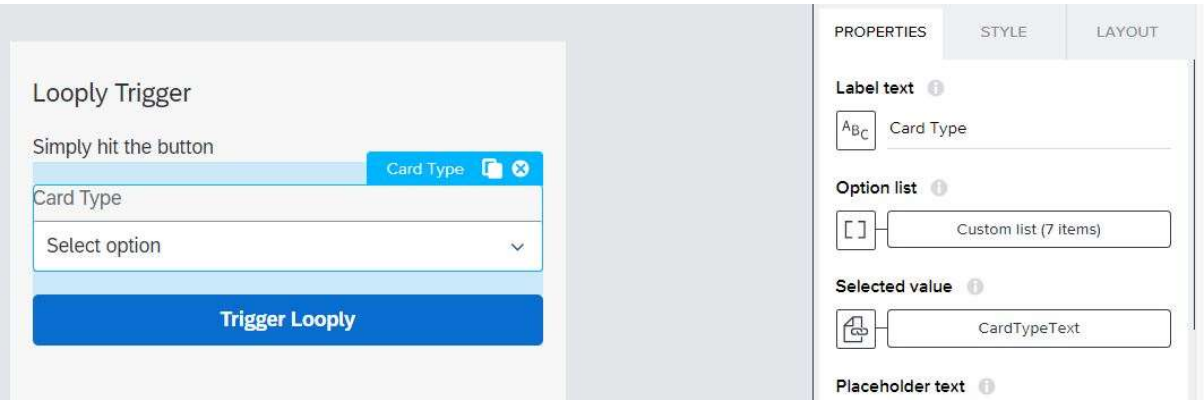
For the SAP Build app design I added a heading, subheading, a drop-down list and a button.



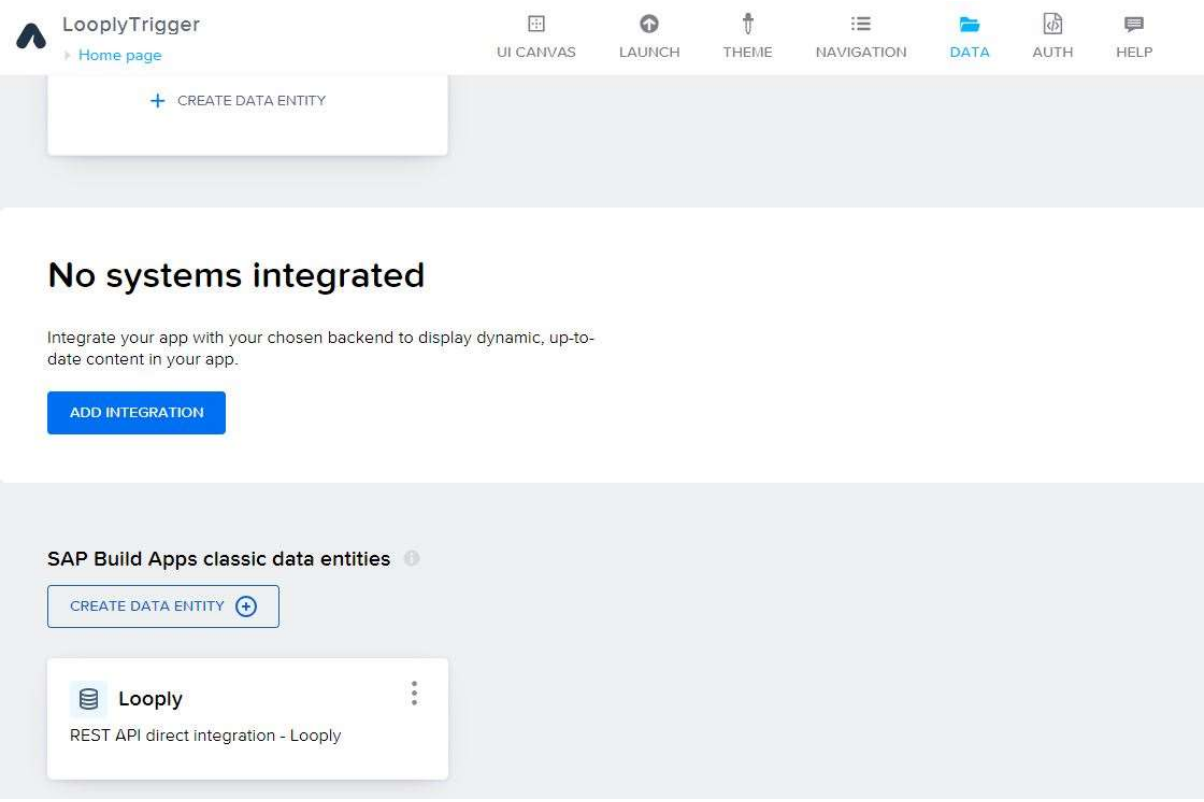
I hardcoded values in the drop-down list, giving each entry a integer value and a readable label that related to the sample cards I built in Looply.



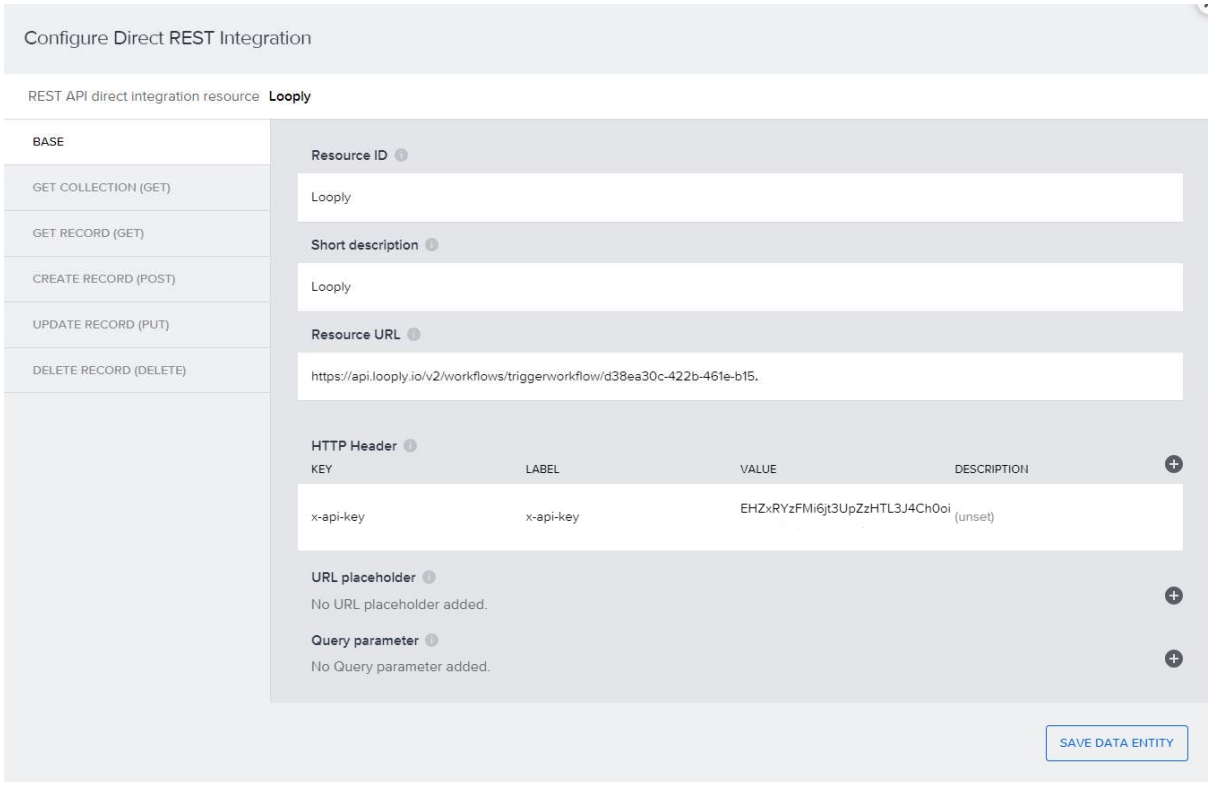
I then created a 'Page variable' called 'CardTypeText' and bound the Selected Value of the drop-down list to the variable.



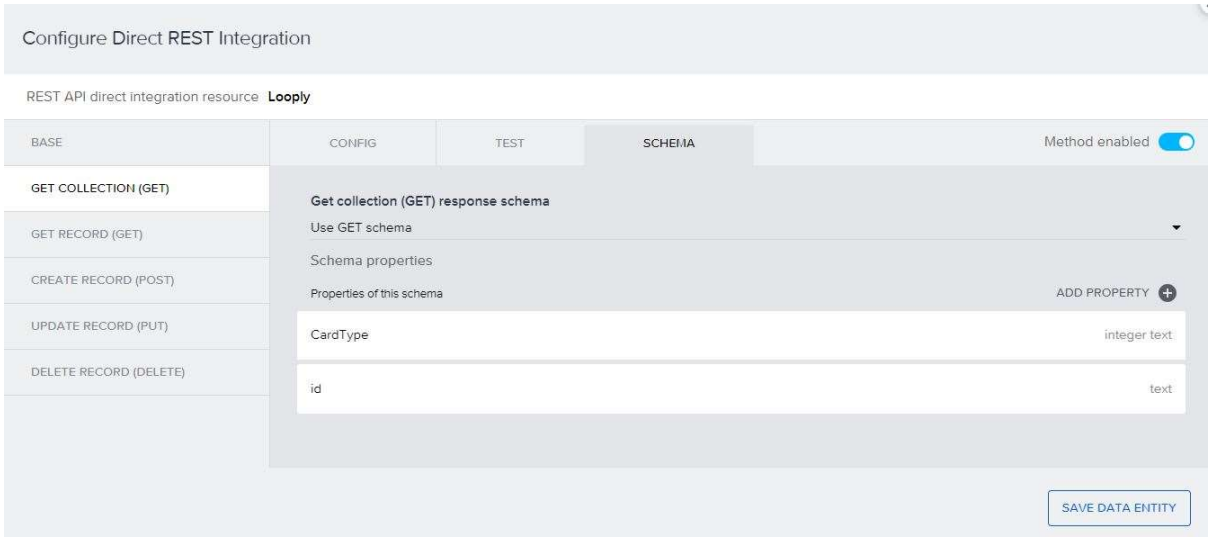
Next I needed to define my data connection to Looply.



I used the SAP Build Apps classic data entities option, and pasted in the 'Request URL' string from the Looply Workflow into the 'Resource URL' field. I also added an HTTP Header to pass in a generated Looply API key.

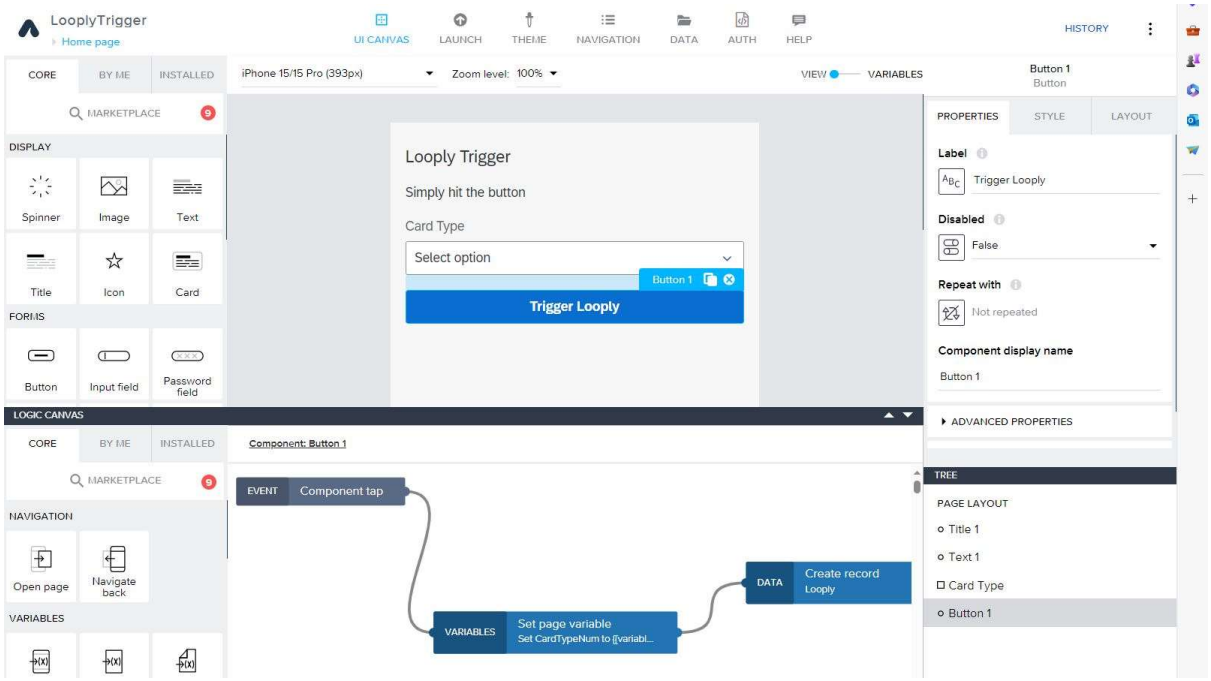


Under ‘Get Collection’ I added a field ‘CardType’ to the schema, and then I could save the data entity.

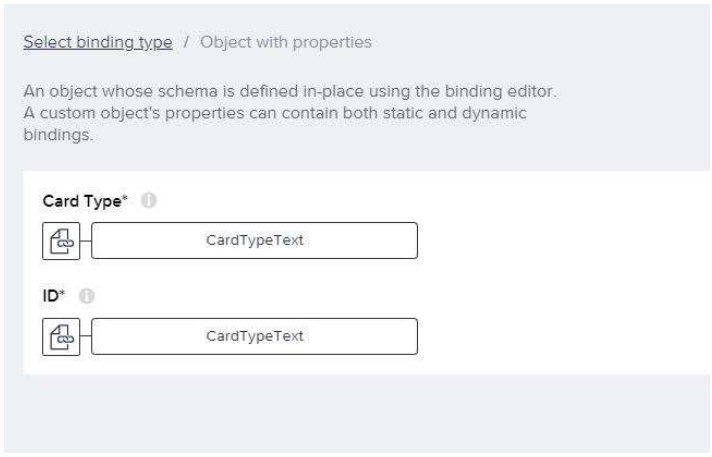


Now, back in the UI Canvas, I could add some logic to the button.

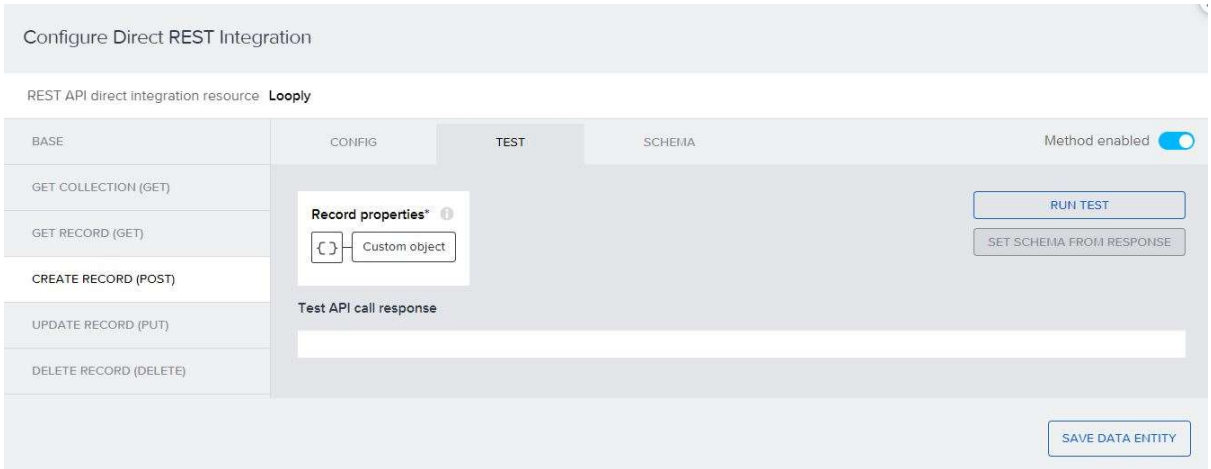
Linking a SAP Build app to Microsoft Teams



Here I passed in the page variable to the data connection schema.

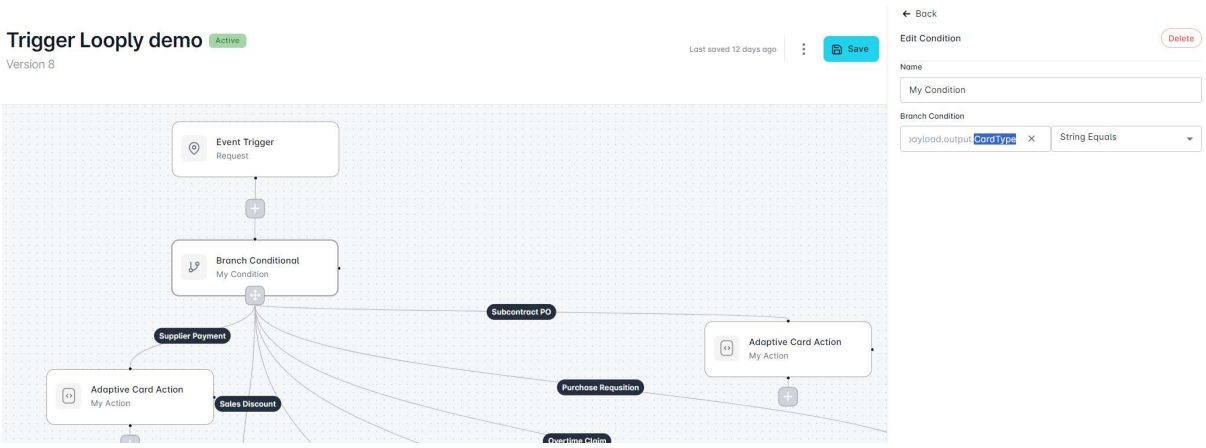


Back in the data connection, I was ready to use the 'RUN TEST' option under 'CREATE RECORD' to try to trigger the Looply Workflow.

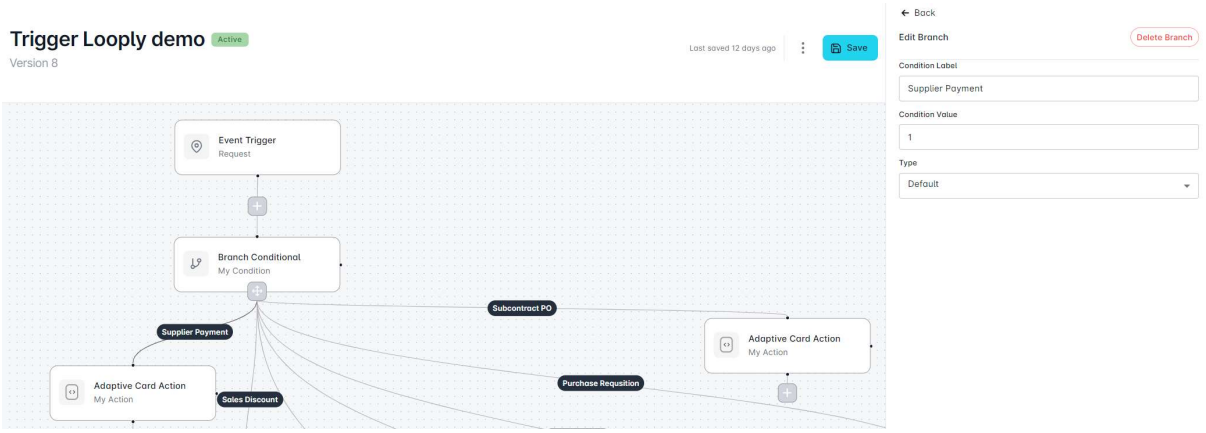


Completing the Looply Workflow

Back in the Looply Workflow, I can now add a ‘branch’ step based on a condition, which checks the value of the ‘CardType’ field.

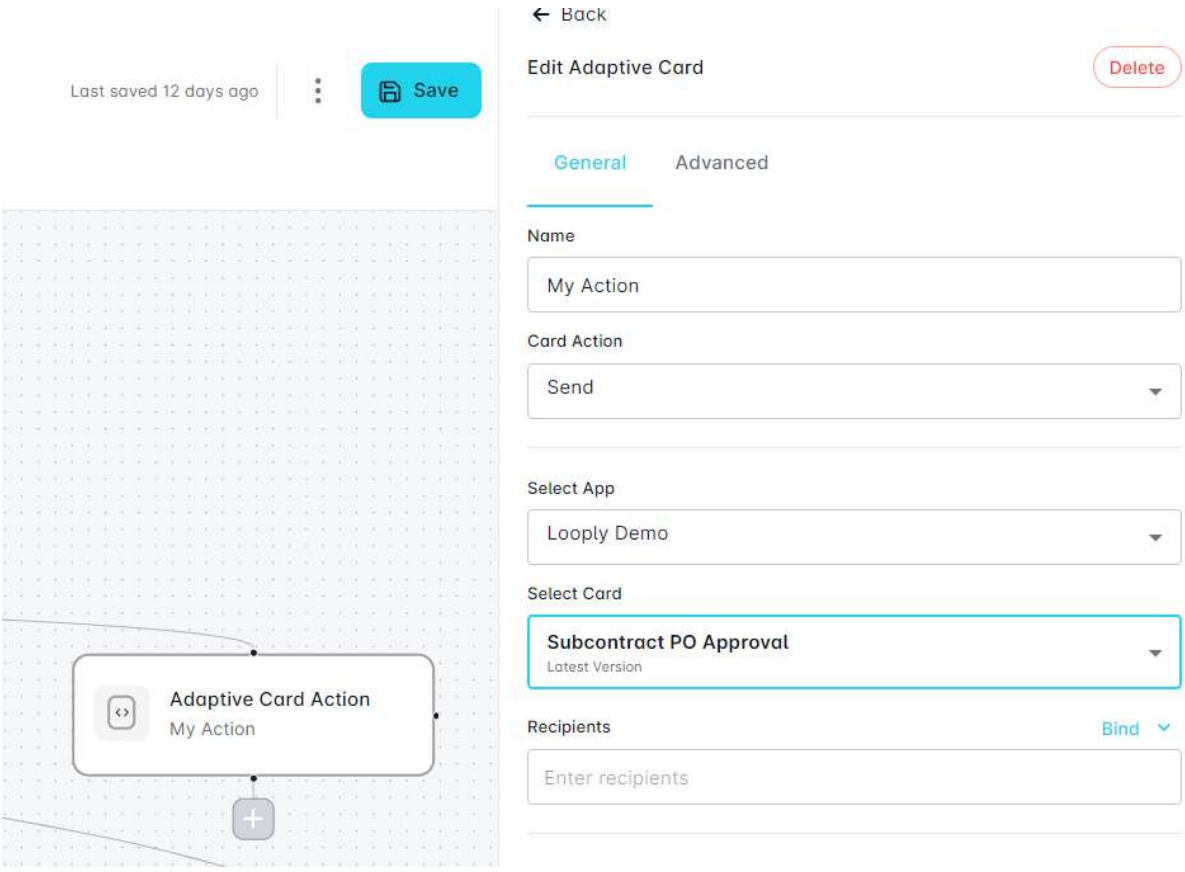


Each branch has a configured value:



At the end of each branch, I trigger a different adaptive card:

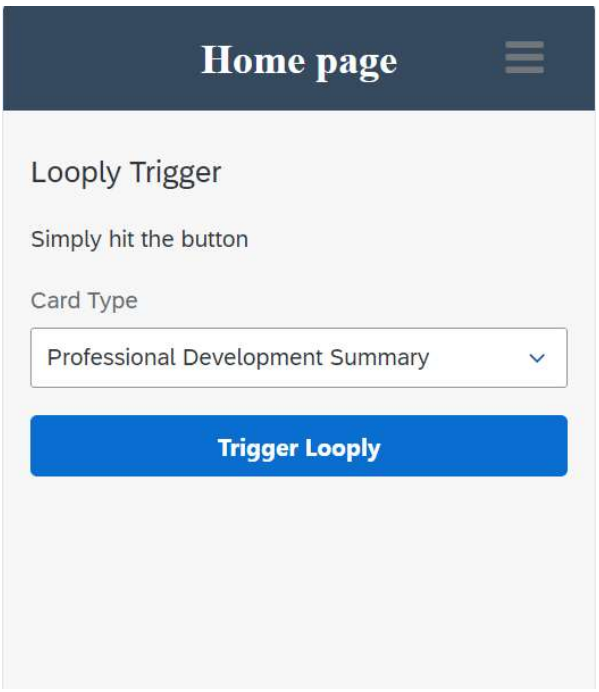
Linking a SAP Build app to Microsoft Teams



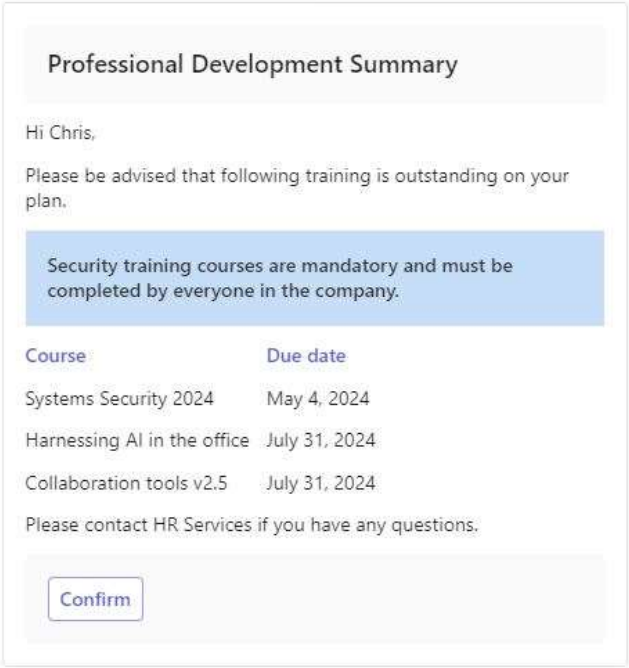
I select an app already created in Looply, and for the purpose of this demo, just hard-code a recipient email address.

Then I'm good to go to test. I can preview the app on desktop or phone:

Select a card type:



Card appears in Teams:



Summary

With no development I was able to build my demo, triggering different Adaptive Cards in Teams based on a simple SAP Build app.

video